REMARKS

[0003]   Applicant respectfully requests reconsideration and allowance of all of the claims of the application.   Claims 1-3, 5-26, 34-38, 40-42, and 45-46 are presently pending. Claims 1, 7-10, 13, 26, 34, 35, 37, 38, 40, and 45 are amended herein. Claims 4 and 39 are cancelled without prejudice or disclaimer.

Statement of Substance of Phone Interview

[0004]   Examiner Stevens graciously talked with me—the undersigned representative for the Applicant—on October 31, 2008. Applicant greatly appreciates the Examiner's willingness to talk.   Such willingness is invaluable to both of us in our common goal of an expedited prosecution of this patent application.

[0005]   During the interview, we discussed how the claims differed from the cited reference, namely Parry. Without conceding the propriety of the rejections and in the interest of expediting prosecution, we also proposed several possible clarifying amendments.

[0006]   Examiner Stevens indicated that he believed although reference Parry does not explicitly teach all the features embodied in some independent and dependent claims, such features were inherent in Parry. We disagreed with Examiner Stevens. Nevertheless, we proposed amendments further highlighting distinctions from the reference. Examiner Stevens indicated such amendments helped to distinguish over reference Parry, and requested the amendments be presented in writing for further consideration.

[0007]   Applicant herein amends the claims in the manner discussed during the interview. Accordingly, Applicant submits that the pending claims are allowable over the cited art of record for at least the reasons discussed during the interview.

Formal Request for an Interview

[0008]   If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.
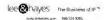
[0009]   Please contact me to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for me, I welcome your call as well. My contact information may be found on the last page of this response.

Claim Amendments

[0010]   Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 7-10, 13, 26, 34, 35, 37, 38, 40, and 45 herein. Claims 4 and 39 are cancelled without prejudice or disclaimer. Applicant amends claims to clarify claimed features. Such amendments are fully supported by the Application, and are made to expedite prosecution and more quickly identify allowable subject matter. Such amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed invention in response to the cited reference.

Substantive Matters

Claim Rejections under § 102(e)

[0011]   Claims 1-26, 34-42, 45 and 46 are rejected under 35 U.S.C. § 102(e) for being anticipated by U.S. Patent No. 6,845,508 to Parry ("Parry"). Applicant respectfully traverses the rejections because, for each rejected claim, no single reference discloses each and every element of that rejected claim.[1] Furthermore, the elements disclosed in the single reference are not arranged in the manner recited by each rejected claim.[2] In light of the amendments presented herein and the discussion during the above-discussed Examiner interview, Applicant submits that these rejections are moot. Accordingly, Applicant asks the Examiner to withdraw these rejections.

[0012]   In rejecting various claims including claim 1, the Office took the position that Parry discloses "intercepting non-native (example of a 32 bit driver that is design to provide common architecture to Windows i.e., Column 4, lines 38-52) program modules, the non-native (example of a 32 bit driver that is designed to provide common architecture to Windows i.e., column 4, lines 38-52; suggestion of a 64 bit bus, column 4, lines 17-21) kernel (column 4, lines 8-12) calls calling a native kernel having access to hardware through one or more device drivers and hardware interfaces native to the native kernel; converting the intercepted non-native (example of a 32 bit driver that is design to provide common architecture to Windows i.e., column 4, lines 38-52) kernel calls into

---

[1] "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989); also see MPEP §2131.
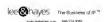
[2] See *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

native kernel calls; and delivering the converted native kernel calls to the native kernel without…" (Office Action 07/02/2008 at pp.4-5).

[0013]   Applicant respectfully traverses the rejections for at least the following reasons.

PCI data bus that supports 32 bit or 64 bit transfer rate has nothing to do with native or non-native kernel calls

[0014]   32-bit and 64-bit and native/non-native code or native/non-native call are two independent and unrelated technical terms. The term "PCI", as defined in Parry, refers to "Peripheral Component Interconnect," a high-performance 32 bit or 64 bit bus designed to be used with devices that have high bandwidth requirements such as the display subsystem. (Parry at col.4, lines 17-21). The term "native code," however, refers to program code that is written to run on a specific processor using that processor's instruction set. (Webopedia, at http://webopedia.com/TERM/N/native_code.html). "Native code," also referred to as "machine code", may be regarded as a primitive programming language or as the lowest-level representation of a compiled or assembled computer program. Programs in interpreted languages (like BASIC, Matlab, etc) or compiled languages (such as C, C++, PASCAL, JAVA, etc) are not "native" programs. These programs must be interpreted or compiled into "native code" for a specific type of CPU. As technology advances nowadays, some compiled languages start taking advantage of 64-bit bus and supporting 64-bit compilation. However, merely supporting 64-bit compilation, or taking advantage of 64-bit bus, does not necessarily turn those programs into "native code." Today's compiled programming language still requires a compiler to compile, or translate, those applications written in compiled language codes
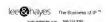
(like C, C++, Java, etc) into "native code" for execution at a specific operating system, even though the compiled programming language supports 64-bit.

[0015] In other words, the 32-bit and 64-bit and native/non-native code or native/non-native call are two independent and unrelated technical terms. Applicant respectfully traverses the rejection in the Office Action to relate 32-bit or 64-bit support to native or non-native calls.

Kernel Mode or Drivers in Parry are focused at hardware level, not at software level as claimed

[0016] "Kernel mode," as defined in Parry, refers to the processor mode which allows full, unprotected access to the system. A driver or thread running in kernel mode has access to system memory and hardware. (Parry, col. 4, lines 8-11). "Driver" is the kernel mode used to either control a hardware device or to emulate a hardware device. (Parry, col. 3, lines 54-56). The term "WDM" in Parry is also focused on hardware. WDM, or Windows 32 Driver Model, is a 32 bit driver model based on the Windows NT driver model that is designed to provide a common architecture of I/O services and binary-compatible device drivers for both Windows NT and Windows operating systems for specific classes of drivers. (Parry, col. 4, lines 38-44).

[0017] Independent claim 1, as amended, recites (in part with emphasis added):

intercepting non-native kernel calls from non-native program modules, _**the non-native kernel calls calling a native kernel of a native operating system**_, _**the non-native kernel calls comprising non-native instructions not operable on the native operating system, the native kernel being part**_

of the native operating system and having access to hardware through one or more device drivers and hardware interfaces native to the native kernel;

converting…

delivering the converted native kernel calls to the native kernel for direct execution at the native kernel of the operating system without ….
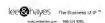
[0018]   In view of the above emphasized features, Applicant respectfully submits that the method for implementing a kernel emulator is more focused on software level than Parry. For example, the "native kernel" is part of a native operating system, which, according to Fig. 3 and associated text, is on top of a hardware level.   Unlike Parry, which discloses a kernel mode where a driver or a thread has access to system memory and hardware, the emulator recited in claim 1 "[delivers] the converted native kernel calls to the native kernel for direct execution at the native kernel of the operating system…"

Many other features as claimed are not disclosed in Parry

[0019]   Claim 1, as amended, recites (in part with emphasis added):

converting the intercepted non-native kernel calls into native kernel calls, the native kernel calls comprising native instructions operable on the native operating system and calling the native kernel of the native operating system, *the converting comprising translating addresses from non-native length within the non-native kernel calls into native length in the native kernel calls*…

[0020]   Applicant respectfully submits that at least the step of "converting" is not disclosed or enabled in Parry.

[0021] In rejecting claim 1, the Office held the position that Parry teaches, *inter alia*, "converting the intercepted non-native (example of a 32 bit driver that is design to provide common architecture to Windows i.e., column 4, lines 38-52) kernel calls into native kernel calls…" (Office Action 07/02/2008 at p.4).

[0022] The WDM, or the Windows 32 Driver Model in Parry, which is designed to provide a common architecture of I/O services and binary-compatible device drivers, is not *per se* designed to perform a conversion.

[0023] Without conceding the propriety of the rejections with respect to "converting" being performed in WDM, Applicant further submits that amended feature, "the converting comprising translating addresses from non-native length within the non-native kernel calls into native length in the native kernel calls," is not disclosed in Parry.

[0024] During the interview, Examiner Stevens indicated that these features, although not explicitly disclosed in Parry, are inherent in Parry. Applicant respectfully disagrees. In order for a feature to be "inherent" the feature must *necessarily* exist in the reference; the claim as a whole must be enabled by the reference.

[0025] Parry teaches drivers in kernel mode to control a hardware device or to emulate a hardware device. Parry further provides WDM to provide a common architecture of I/O services and binary-compatible device drivers. This is not what is claimed.

[0026] In particular, amended claim 1 recites "intercepting non-native kernel calls from non-native program modules… the native kernel being part of the native operating system… converting the intercepted non-native kernel calls into native kernel calls… delivering the converted native kernel calls to the native kernel for direct execution at the

native kernel of the operating system." Even if the WDM in Parry may involve a data conversion as Examiner Stevens suggested, such conversion is not *necessarily* performed in the same way as claimed.

[0027]   A person having ordinary skill in the art ("PHOSITA") at the time of the instant invention, after reading Parry, would not know whether a conversion is performed in WDM in Parry, and even if it is, as suggested by Examiner Stevens, how such conversion would be conducted in WDM. For example, the act of converting comprising "translating address from non-native length within the non-native kernel calls into native length in the native kernel calls" is not disclosed in Parry.

[0028]   Other features related to the act of "converting" in dependent claims are not disclosed or enabled in Parry. For example:

- "the converting further comprises translating a non-native paradigm for passing parameters into a native paradigm for passing parameters" as recited in claim 2;
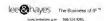
- "the converting further comprises translating non-native CPU instructions into native CPU instructions" as recited in claim 3;

- "the converting further comprises converting non-native argument format into native argument format" as recited in claim 5;

- "the converting further comprises translating words from non-native word size into native word size" as recited in claim 6;

are not explicitly disclosed or enabled in Parry.

[0029]   Furthermore, claimed features in claims 7-10, i.e.,:

- "limiting addressable memory to a range addressable by non-native program modules",

- "managing memory space that is accessible to both native and non-native program modules",

- "synchronizing a native shared data structure with a non-native shared data structure",

- "managing memory space accessible to both native and non-native program modules; and mapping versions of process shared data structures (process SDSs) and versions of thread shared data structures (thread SDSs) between native and the non-native program modules",

are all absent in Parry.

[0030]   Therefore, independent claim 1, as amended, is respectfully asserted patentably distinct from Parry. Additionally, dependent claims 2, 3, 5-10 are patentably distinct from Parry for additional and independent reasons mentioned above.

[0031]   Independent claims 13, 26, 34, 38, 40 and 45 are amended to incorporate features similar to claim 1, and therefore are also asserted patentably distinct from Parry for similar reasons as claim 1, discussed above.

[0032]   For example, regarding claims 13 and 34, it is _not necessarily the case_ that Parry performs each of the claimed acts of the methods in the same way claimed, in part because Parry does not _necessarily_ perform the acts as recited in these claims at all. Thus Applicant submits that the disclosure of Parry fails to anticipate these claims at least because it does not disclose each and every act as set forth in the claims, either expressly

or inherently, as recited in claims 13 and 34. Accordingly, Applicant respectfully requests that the rejection of these claims be withdrawn and the claims allowed.

Dependent Claims 11, 12, 14-25, 35-37, 41, 42, and 46

[0033]   These claims ultimately depend upon independent claims 1, 13, 26, 34, 40 and 45. As discussed above, claims 1, 13, 26, 34, 40 and 45 are allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Conclusion

[0034]   All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action**. Please call or email me at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC
Representatives for Applicant

/Ningning Xu Reg. No. L0293/   Dated: 2008-11-03
Ningning Xu (ningning@leehayes.com; (509) 944-4726)
Registration No. L0293

Bea Koempel-Thomas (bea@leehayes.com; (509) 944-4759)
Registration No. 58,213

Customer No. **22801**

Telephone: (509) 324-9256
Facsimile: (509) 323-8979
www.leehayes.com

Serial No.: 09/847,535
Atty Docket No.: MS1-0665US
Atty/Agent: Ningning Xu
-26-
lee & hayes   The Business of IP ™
www.leehayes.com   509 324 9256